CLAIMS

What is claimed is:

1. A method for thread scheduling for a multithreaded application in a multiprocessor system, comprising:

obtaining information on data sharing behavior among multiple threads of said multithreaded application;

grouping said multiple threads into at least one group based at least in part on said information on data sharing behavior among said multiple threads; and

scheduling said group of threads to target processors, said target processors being in proximity to each other electronically.

2. The method of claim 1, wherein obtaining information on data sharing behavior comprises:

compiling said multithreaded application;

analyzing data sharing behavior among said multiple threads; and

collecting said information on data sharing behavior.

3. The method of claim 1, wherein grouping said multiple threads comprises:

identifying tightly coupled threads; and

placing said identified threads into the same group.

4. The method of claim 1, wherein scheduling said group of threads to target processors comprises:

assigning threads in the same group to said target processors that are in a cluster; and

if there are not enough available processors in said cluster, assigning the rest of the treads to processors that are electronically in proximity to said cluster.

5. The method of claim 1, further comprising:

executing said multiple threads of said multithreaded application on said multiprocessor system; and

observing data sharing behavior among said multiple threads during execution.

6. The method of claim 5, further comprising:

providing feedback of said data sharing behavior to regroup said multiple threads; and

rescheduling, at the next scheduling time, said regrouped multiple threads by assigning threads in the same group to target processors that are electronically in proximity to each other.

7. The method of claim 1, wherein grouping said multiple threads further comprises regrouping said multiple threads based on feedback of data sharing behavior among said multiple threads during execution.

8. An article comprising a machine-readable medium that contains instructions, which when executed by a processing platform, cause said processing platform to perform operations comprising:

obtaining information on data sharing behavior among multiple threads of said multithreaded application;

grouping said multiple threads into at least one group based at least in part on said information on data sharing behavior among said multiple threads; and

scheduling said group of threads to target processors, said target processors being in proximity to each other electronically.

9. The article of claim 8, wherein obtaining information on data sharing behavior comprises:

compiling said multithreaded application;

analyzing data sharing behavior among said multiple threads; and

collecting said information on data sharing behavior.

10. The article of claim 8, wherein grouping said multiple threads comprises:

identifying tightly coupled threads; and

placing said identified threads into the same group.

11. The article of claim 8, wherein scheduling said group of threads to target processors comprises:

assigning threads in the same group to said target processors that are in a cluster; and

if there are not enough available processors in said cluster, assigning the rest of the threads to processors that are electronically in proximity to said cluster.

12. The article of claim 8, wherein said operations further comprises:

executing said multiple threads of said multithreaded application on said multiprocessor system;

observing data sharing behavior among said multiple threads during execution;

providing feedback of said data sharing behavior to regroup said multiple threads; and

rescheduling, at the next scheduling time, said regrouped multiple threads by assigning threads in the same group to target processors that are electronically in proximity to each other.

13. The article of claim 8, wherein grouping said multiple threads further comprises regrouping said multiple threads based on feedback of data sharing behavior among said multiple threads during execution.

14. An apparatus for thread scheduling for a multithreaded application on a multiprocessor system, comprising:

a compiler to compile said multithreaded application, to analyze data sharing behavior among multiple threads of said multithreaded application, and to obtain information on data sharing behavior among said multiple threads; and

a thread scheduler to receive said information on data sharing behavior among said multiple threads and to schedule said multiple threads to target processors based at least in part on said information on data sharing behavior.

15. The apparatus of claim 14, wherein said thread scheduler groups tightly coupled threads into the same group and assigns said threads in the same group to processors in a cluster, and if there are not enough available processors in said cluster, assigns the rest of said treads to processors that are electronically in proximity to said cluster.

16. The apparatus of claim 14, further comprises a plurality of processors to receive and execute said scheduled threads.

17. The apparatus of claim 16, further comprises a feedback module to observe data sharing behavior among said scheduled threads during execution and to provide feedback on said data sharing behavior during execution to said thread scheduler.

18. The apparatus of claim 17, wherein said thread scheduler regroups and reschedules, at the next scheduling time, said multiple threads to target processors based on said feedback on said data sharing behavior during execution.

19. A multiprocessor system, comprising:

a synchronous dynamic random access memory ("SDRAM");

a plurality of processors coupled to access said SDRAM via a system interconnect, said plurality of processors forming at least one cluster, wherein processors in the same cluster have a shared storage device;

at least one compiler in one or more of said plurality of processors to receive and compile a multithreaded application, to analyze data sharing behavior among multiple threads of said multithreaded application, and to obtain information on data sharing behavior among said multiple threads; and

at least one thread scheduler in one or more of said plurality of processors to receive said information on data sharing behavior among said multiple threads from said compiler, and to schedule said multiple threads to said plurality of processors based at least in part on said information on data sharing behavior.

20. The system of claim 19, wherein said thread scheduler groups tightly coupled threads into the same group and assigns said threads in the same group to processors in a cluster, and if there are not enough available

processors in said cluster, assigns the rest of said treads to processors that are electronically in proximity to said cluster.

21. The system of claim 19, further comprises a feedback module to observe data sharing behavior among said scheduled threads during execution and to provide feedback on said data sharing behavior during execution to said thread scheduler.

22. The system of claim 21, wherein said thread scheduler regroups and reschedules, at the next scheduling time, said multiple threads to said plurality of processors based on said feedback on said data sharing behavior during execution.